

# MULTISPECTRAL HIGH FIDELITY RADAR SCENE GENERATOR

## FINAL REPORT SBIR PHASE I; TOPIC N99-059

MRA Document No. P376-Final

**For Period:** Feb 12 – Aug 12, 1999  
**Date Prepared:** September 2, 1999  
**Contract No.:** N68335-99-C-0126

**Prepared for:**

Government Technical Liaison  
Naval Air Systems Command, Headquarters  
Attn: Mr. James Evans, Bldg. 2272  
Suite 535, Code PMA 272  
Patuxent River, MD 20670

**Prepared by:**

Anand Kelkar  
Malibu Research Associates  
26670 Agoura Rd.  
Calabasas, CA 91302

**Distribution Statement A: Approved for public release; distribution is unlimited.**



### Malibu Research

26670 Agoura Road  
Calabasas, CA 91302-1974  
(818) 880-5494 Fax 880-5499

DTIC QUALITY INSPECTED 4

19990909 054

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 990902	3. REPORT TYPE AND DATES COVERED Final (990212 through 990812)	
4. TITLE AND SUBTITLE Multispectral High Fidelity Radar Scene Generator			5. FUNDING NUMBERS C: N68335-99-C-0126	
6. AUTHOR(S) Anand Kelkar				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Malibu Research Associates, Inc. 26670 Agoura Road Calabasas, CA 91302-1974			8. PERFORMING ORGANIZATION REPORT NUMBER P376-Final	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Air Systems Command, Headquarters Attn: Mr. James Evans, Bldg. 2272 Suite 535, Code PMA 272 Patuxtent River, MD 20670			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT "Approved for public release; distribution is unlimited."			12b. DISTRIBUTION CODE A	
13. ABSTRACT (Maximum 200 words)  <p>The product approach developed herein has excellent potential as a training tool for various RADARs. It provides an operator interface with the actual on site RADAR hardware. There is some minimum modification that is done to the RADAR system to accept the RSG. Due to the flexibility of the RSG output, RADAR signal injection can be made anywhere from the RF front end (after the 1<sup>st</sup> LNA), through various points in the IF processing chain, all the way to selected points in the digital processing.</p> <p>Additionally, The RSG serves as a performance verification tool for RADAR hardware as it lives out its life cycle. As the hardware or operational software receives upgrades, the RSG can be applied as a final checkout prior to live engagement. The benefit of using the RSG is that the RADAR environment is electronically synthesized, and therefore is exactly repeatable. The RADAR performance improvement then can be assessed objectively as the RADAR goes through its upgrade cycles.</p> <p>The Multispectral High Fidelity RADAR Scene Generator represents a portable and comprehensive test and training environment for the Navy's RADAR assets. Additionally, the option of providing a RADAR evaluation service for other RADAR sponsors becomes available, where the Navy can maintain a center that could assist other RADAR users in the evaluation of RADAR assets and training of operators.</p>				
14. SUBJECT TERMS			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR	

This is the final report for the Multispectral High Fidelity RADAR Scene Generator SBIR (Phase I), Contract # N68355-99-C-0126.

## **1.0 Introduction**

This report covers effort from Feb. 12, 1999 to Aug. 12, 1999 and describes the design of a physics based virtual RADAR Scene Generator (RSG). Though the RSG product is designed keeping in mind its application as a detailed testing and training device for RADAR operators, the Malibu Research implementation is a faithful reproduction of RADAR phenomenology, and consequently, the RSG serves as a design/development tool for RADARs in general, allowing quick iteration and optimization of the signal processor and operational algorithms during RADAR system development and upgrade cycles.

The RSG design relies on currently available COTS hardware and therefore can be built from available components. The state of the COTS art allows parallel processing using multiple processing nodes and is scalable depending on the specific requirement, which is determined by the capacity of the target RADAR signal processor.

An overall design for the RSG has been presented herein. This final report draws from various progress reports that describe the significant design details. These have been attached for reference.

## **1.1 Background**

Malibu Research has produced several real-time physics based target environment generators in the past. These interactive RADAR Environment Simulator (RES) products provide prioritized target fidelity and contained statistically accurate yet generic bulk clutter models as a test of the Doppler fidelity and clutter rejection performance of the RADAR processor. These early RES products have provided a wealth of experience to draw from, providing insight into efficient implementation schemes for complex target amplitude and spectral signatures.

## **1.2 Challenges**

This SBIR program adds several new aspects to the proven RES principles:

- 1) The clutter backscatter information for this SBIR application is required to originate from available DTED and/or DFAD databases. This means that the RADAR has to be virtually placed on the digital database and the backscatter coefficients have to be computed from the actual aspect angle of the local clutter patch to the RADAR line-of-sight.
- 2) A generic model for sky clutter has to be devised. In the past RES products, the sky clutter consisted of multiple point targets (called angels), and sky clutter was a combination of a plethora of this class of targets.

- 3) All User interface issues have to be considered carefully. The past RES products have had engineering interfaces that were command line based, and therefore cumbersome. One of the intended applications for this SBIR is for RADAR operator training. To this end, the user interface must be easy to use and intuitive.

### 1.3 Considerations

Commercial-Off-The-Shelf (COTS) products were considered heavily in the design of the Multispectral RSG. The advantage of COTS hardware is clear in that the development, distribution and technical support of hardware and software is funded by commercial technology and matures much more rapidly than for custom parts. The other side of the coin is that when problems are encountered, the resolution often requires more than one vendor (hardware & software for example), and the high volume problems get the attention, but the single application bug that might be uncovered often gets resolved on a schedule that is convenient to the vendor, not the user.

## 2.0 RSG Design

The High Fidelity Multispectral Scene Generator SBIR work has culminated in a design approach that will provide a flexible interface to existing RADAR hardware while supporting the latest in database model literature. Figure 1 shows a block diagram of this design.

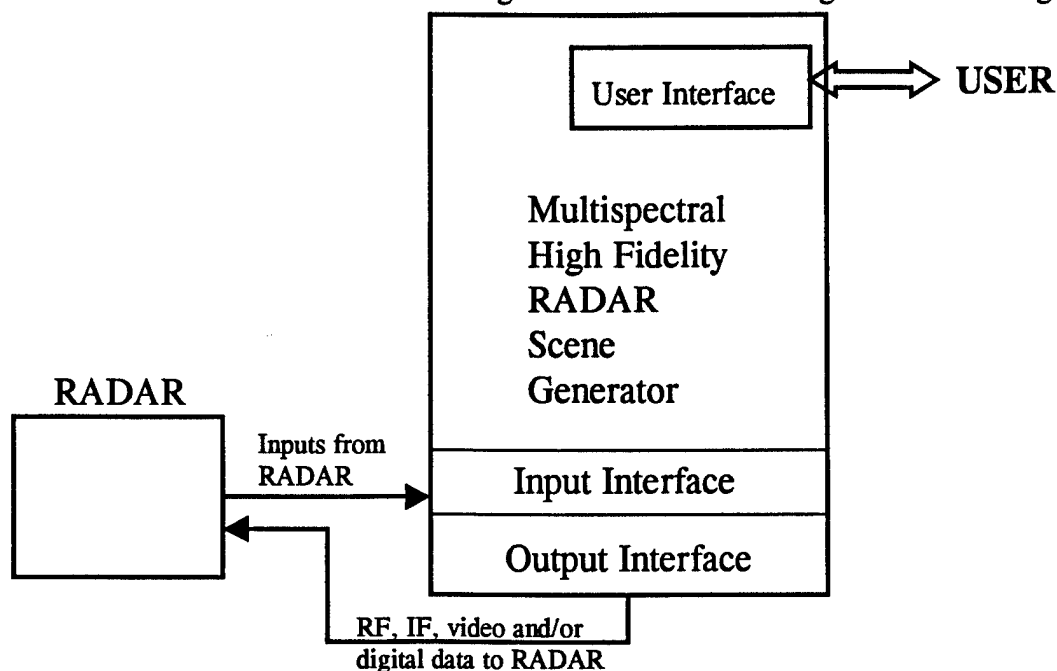


Figure 1 - High Fidelity Multispectral RADAR Scene Generator Hardware Block Diagram

This approach allows the RSG to be applied to a variety of RADAR hardware. With each application, the hardware changes are limited to the RADAR input and output interfaces. Details of the RADAR interface requirements are discussed in the third monthly progress report (attached) covering work from Apr. 12 to May. 12, 1999.

### 3.0 Typical RADAR Application (Requirements)

In the first progress report, a straw man RADAR was discussed. This fictitious RADAR was used as the mark that the SBIR effort had to address. Table 1 below summarizes the requirements presented in the first progress report (attached), covering work done from Feb.12 to Mar.12, 1999. The values listed for the various parameters are representative of state-of-the-art RADARs.

1	Search range:	300	Km
2	Range resolution in search	150	mtr. (1 $\mu$ s)
3	Track gate size	25	uSec
4	Track gate resolution	15	mtr (0.1 $\mu$ s)
5	Azimuth coverage	360°	Steerable
6	Elevation coverage	30°	Steerable
7	Number of simultaneous targets	100	max
8	Number of targets in beam	20	max
9	Azimuth Beamwidth	2°	
10	Elevation Beamwidth	2°	
11	Maximum instrumented range	2000	cells (in search mode)
12	Finest Doppler resolution	1	Hz (0.05 m/sec at 3.0 GHz RADAR freq)
13	Max. unambiguous Doppler	10	KHz
14	Min phenomenologic freq	0.5	Hz clutter motion bandwidth

The COTS processor hardware available today is able to support the data rates implied by the table, in some cases by using a parallel processing approach. This allows all the RADAR specific processing issues to be addressed by the software in the RSG, with minimal hardware used to perform only the interface functions.

### 4.0 Hardware configuration

The RSG application to a specific RADAR is determined by the RADAR processor. The architecture of the RSG is fully scalable and is comprised of several "channels" each capable of supporting the RADAR requirements discussed above.

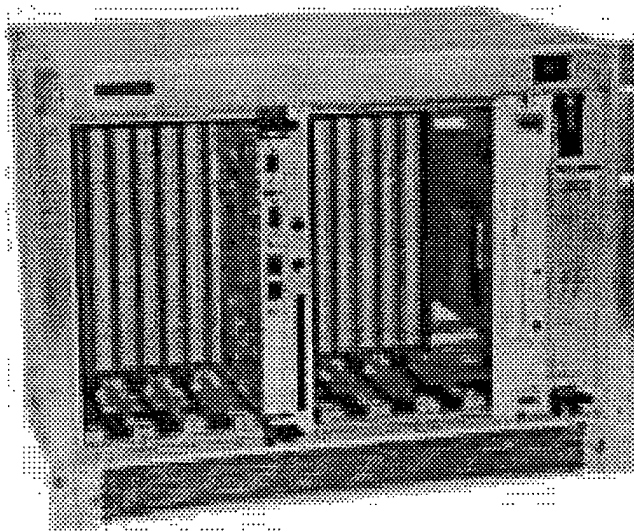
There are basic compute elements that provide the user interface function and the RADAR coordination function, which are common to all RADAR applications. This makes intuitive sense since any scene run by any one operator can be used against a variety of RADARs. Following, there are several channels of DSP engines. These DSP engines perform the signal processing functions to transform the scene into the signals that the RADAR likes to see, using the principles discussed in the first progress report (attached), covering

progress from Feb.12 to Mar.12. The specific RADAR's signal processing capabilities determine how many of these channels are required.

The following paragraphs describe a preferred processing architecture which is more fully described in the third progress report (attached), which covers progress from Apr. 12 to May 12.

#### **4.1 Bus Infrastructure**

The emerging bus standard that more and more COTS suppliers are supporting appears to be Compact PCI (cPCI). This architecture blends the robust and long-lived VME mechanical interface with the electrically simple PCI interface developed through the home computer market. The cPCI chassis is available through various vendors. Our survey of the market indicated that the best supported solution was provided by a local (southern California) distribution company. A photo of a cPCI chassis from this company is shown below. The details of the selection are provided in the third progress report (attached).



**Figure 2 - cPCI chassis planned for the RSG (shown with PC installed)**

#### **4.2 Computer resources**

The user interface function will be handled by a standard PC clone. The currently available Pentium II - 300 MHz processor hardware can easily handle the interface service needs that are planned for the user interface function. The scene manipulation task is relegated to a Power PC (PPC) processor "daughter" card supported on one of the internal buses available on the PC clone. This internal bus supports the full PCI protocol, but is mechanically arranged so that the "daughter" card forms a mezzanine. Hence, the interface is appropriately called the PCI Mezzanine Card (PMC) interface and is supported by the COTS world.

The illustration below shows the cPCI Pentium II processor and the PMC Power PC processor which will support the RSG function. Both processor modules are shown approximately to scale with respect to each other. Note the two white PMC connectors at the lower right corner of the PMC module, which plug into corresponding white sockets on the upper right hand side of the cPCI PC Processor card.

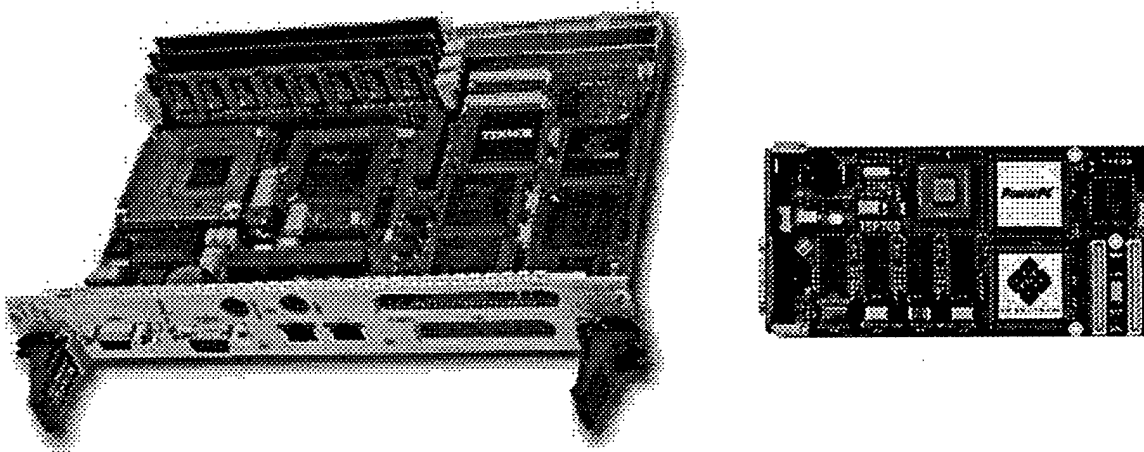


Figure 3 – cPCI Pentium II and PMC Power PC computer resources for the RSG

#### 4.3 Signal Processing Resources

The two processing elements described above provide the user interface and the scene processor, which act during RSG operation to select the pertinent model information that will be viewed by the RADAR at a particular instant in time. This selection is based on the RADAR information received across the RADAR input interface at the RSG.

The model information then is translated into RADAR signals by the DSP hardware in the RSG. This DSP hardware resides on COTS boards that are provided by several vendors. The basic DSP engine that will be used is the Texas Instruments TMS320-C6201 signal processor. Various iterations of this signal processor have been in existence for the past decade.

Up to four DSP chips are supported on cPCI cards available today. The TI TMS320-C6201 was chosen as the potential solution because it was the fastest processor available on the market today. A single “C62” processor has the capacity to support the RADAR requirements identified earlier. Two RADAR channels will be supported by a single quad C62 card, which allows 100% headroom for expansion.

The C62 solution chosen for the RSG is the “Barcelona”, which is supplied by Spectrum Signal Processing of Burnaby, BC in Canada. Details about this card are provided in the third progress report (attached). The Barcelona is shown below:

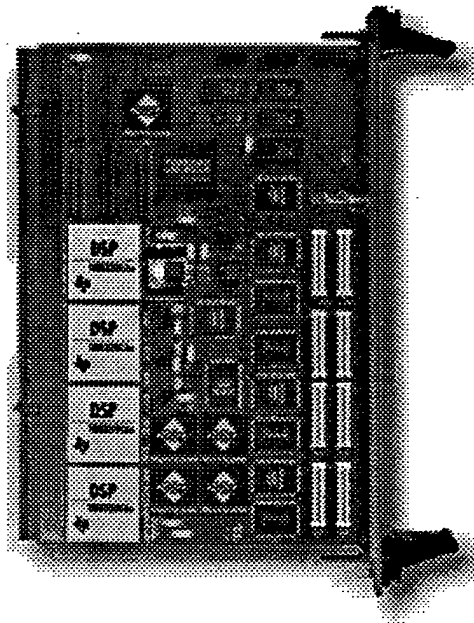


Figure 4 – Quad Texas Instruments TMS320-C6201 chips on a “Barcelona” board

## **4.4 Interface to the RADAR**

### **4.4.1 Inputs to RSG**

The RADAR input interface (from the RADAR to the RSG) is allows the RSG to understand what the current RADAR function is and re-act accordingly. A CPLD (Complex Programmable Logic Device) based interface board will be used to perform this function. The use of the CPLD allows for flexibility in the acquisition and format conversion of signal as received from the RADAR.

The functions of this board include:

- 1) Accepting the RADAR information in serial or parallel (RADAR specific) format.
- 2) Broadcasting the information to the various processors in the implementation chain through an on board serial port available on each TMS320 DSP chip.
- 3) Acknowledging receipt of the RADAR input data (if required).
- 4) Buffering data for additional “fast response” signal applications that the RADAR might require

These interface functions will be performed on the “Digital I/O Board”.



#### 4.4.2 Outputs from the RSG

The handshaking functions associated with RADAR-RSG data transfer are handled on the Digital I/O Board described above. The "Live" data that the RADAR expects to see are synthesized in digital form by the quad DSP "Barcelona" boards and then translated to analog (IF) signals by the Digital-to-IF Board.

This board accepts A/D clocking and IF phase reference signals from the RADAR and applies the digitally synthesized target data to modulate the IF reference and create the desired "Live" data at IF frequencies. Shown below is a block diagram of this process as implemented for the US Army's Firefinder RES.

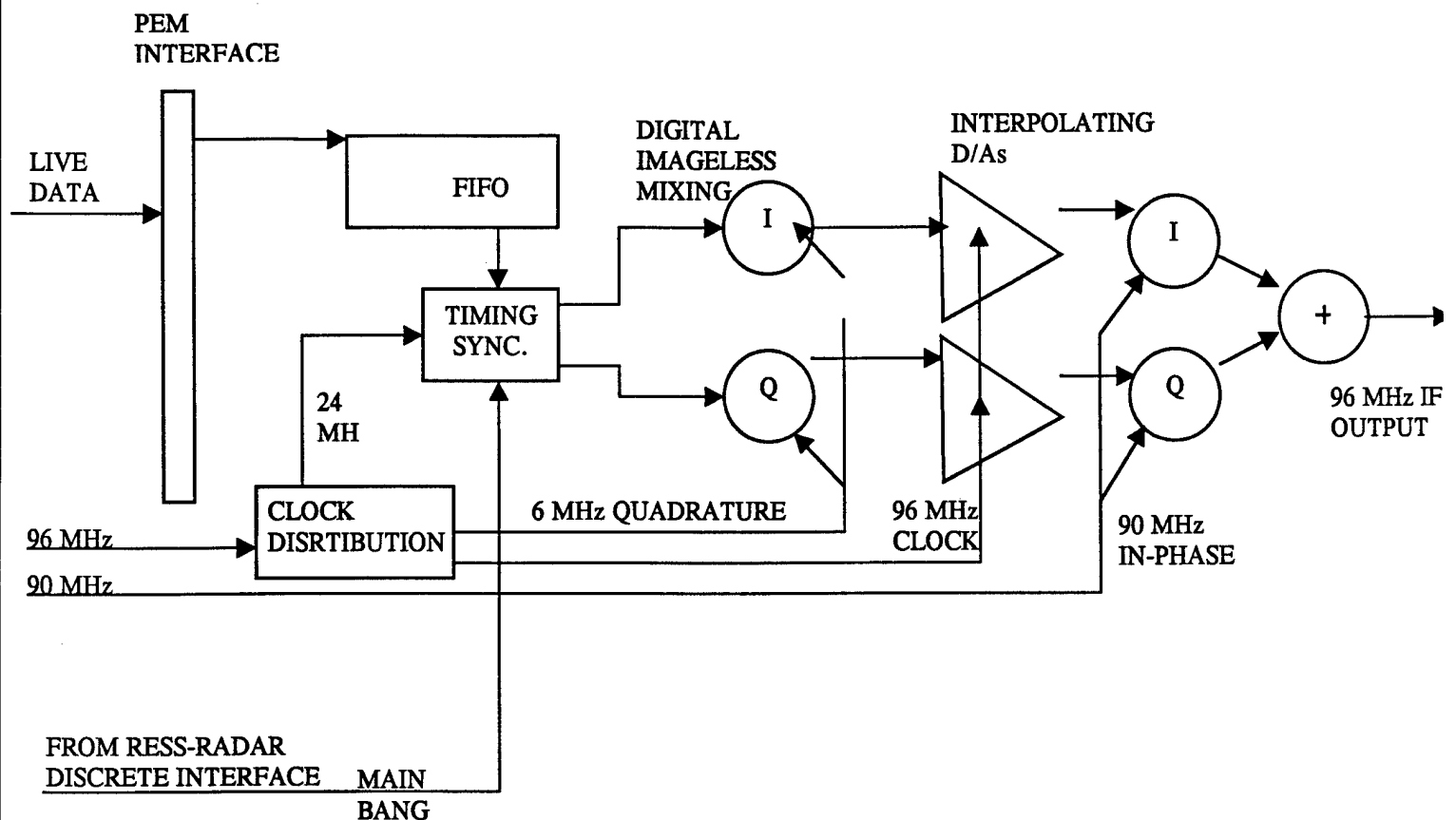


Figure 5 - Digital-to-IF conversion module from a RES for the Firefinder Army RADAR

This Digital-to-IF module is designed to provide the RADAR signals at an IF frequency. With each RADAR application, the IF frequency is expected to vary. When injection at RF is required, the same block diagram describes the functions. The difference is in the RADAR reference. In certain situations, the IF "Live" signals may have to be synthesized at IF and block upconverted to the RADAR RF frequency before application.

Note the use of a "PEM" interface in the block diagram. This is a 400 MB/s interface that is provided on the "Barcelona" board and allows high speed buffered data transfers from any of the four DSP chips on the board.

## **5.0 Database**

Proper operation of the RSG relies heavily on an accurate model database. To properly account for various physical phenomena and their effect on the RADAR propagation, and therefore, RADAR operation, processing threads were created to describe the overall output. The second progress report, covering work from Mar. 12 to Apr. 12 (attached) describes these processes in detail.

The RSG will contain an off-line processing element that allows the user to create a RADAR environment of choice by:

- 1) Loading a specific DTED CD and place the RADAR on the virtual earth.
- 2) Creating target scenarios by choosing targets as desired from a palette of available possibilities.
- 3) Creating clutter scenarios by defining different clutter entities and weather profiles (described in the second progress report; Mar. 12 to Apr. 12).
- 4) Combining scenarios pre-defined through the steps above.

The output of this off-line process will generate a "run-time" file that will be "played" by the RSG in synchrony with the RADAR's operation and will contain the database components that originate from the models. In generating the database, the following will be taken into consideration:

- 1) RF refraction... the 4/3 earth model is the most simple and widely used, and will be the default model.
- 2) Coordinate transform – from absolute position on the virtual (DTED) surface to the RADAR antenna relative R/Az/EI space. The WGS-84 transform model will be used along with other trigonometric conversions.
- 3) RF propagation due to ducting and weather and earth boundary conditions. This behavior is predicted by applying the Parabolic Wave Equation and is discussed in the second progress report (attached)
- 4) Clutter backscatter variation due to weather influences. These factors stem from a change in the reflective characteristics of the terrain as well as from the backscatter contributions of precipitation.
- 5) Spectral contributions from wind blown clutter as well as from swaying vegetation due to wind.

The second progress report details these and other contributors to the clutter generation process.

## **6.0 RSG Development Environment(s) and Operating Systems**

As a part of this study, we have not only identified COTS/NDI hardware candidates, but have also investigated development and operating systems, which can integrate the COTS hardware into a fully functioning RSG. In doing this investigation, we were able to utilize experience gained from our development of current FIREFINDER RES products.

### **6.1 RSG User Interface Operating Software**

As noted above in Paragraph 4.2, the User Interface of the RSG is hosted on a Pentium-II PC installed in the compact PCI (cPCI) chassis. The operating system chosen for this processor is Windows NT, a 32-bit true multi-tasking, virtual-memory operating system with widespread availability of software-development products (e.g., from DSP board vendors) as well as from Office Automation products (e.g., Office97 and the Netscape web browser). Version 4.0 of NT is the current release, with Service Pack 4 updates providing cumulative bug fixes and new features. It is quite similar to the even more popular Windows95 and Windows98 environments, but is more robust and advanced. Code written for the Win32 application programming interface (API) runs on 95, 98 and NT, but there are significant differences in hardware support (e.g., for DSP devices used in the RES, which are supported only under NT).

The RSG User Interface can be written in any combination of Microsoft Visual C, Visual C++ and Visual Basic, as desired by the programmers, to take advantage of graphical features and ease of development. The vendor-supplied DSP application libraries can be linked to any of these languages via Microsoft's Dynamic Linked Library (DLL) approach. However, Windows is not a real-time operating system – for example, an interrupt-handling time of 100  $\mu$ s is typical with Windows, while times of 1-5  $\mu$ s are feasible with real-time operating systems discussed below. If required, a real-time version of Windows is available from third-party sources (at significant expense).

### **6.2 RSG Real-time Computation Modules**

The real-time computation in the RSG breaks into two main parts: Dwell Processing and Pulse Processing. The first of these can best be achieved either by a Digital Signal Processor (DSP) or by a general-purpose processor, while (in our architecture) the latter requires the capabilities of the DSP. Dwell processing occurs once per dwell (whose minimum length is about 1.5 ms), while Pulse Processing needs to occur at the pulse-repetition interval (whose minimum length is about 100  $\mu$ s). At present, our design has allocated the Dwell Processing to a PowerPC module, which is much faster than the C6201 DSP for general (including floating-point) computation. However, it is possible that the soon-to-be-available C6701 floating-point DSP may be able to achieve the Dwell Processing more efficiently than the PowerPC module, thus potentially simplifying our architecture.

#### **6.2.1 DSP Modules**

The DSP modules in our architecture have been selected from one of the leading DSP vendors, Spectrum Signal Processing (SSP). SSP has a long history with both Analog Devices

(SHARC) and Texas Instruments (C4x, C6x) DSPs, and we believe their products offer significant advantages to the RSG architecture. We are specifically using a dual-processor C6201 PCI board (the Daytona) for development purposes, and are planning to deliver a quad-processor C6201 cPCI board (the Barcelona) in the final RESS. For development purposes, the Daytona represents a single RSG channel (of the 6 initially required), and the Barcelona thus achieves integration of two RSG channels on a single 6U size cPCI board. Both boards have identical hardware architecture in terms of memory and inter-processor communication devices, and thus run the same release of software, described by SSP as the "Fast Track" environment.

Software for the Fast Track environment consists of an NT driver DLL together with library software and tools to enable development of "host" (Windows) and "target" (DSP) code. The host code is built using standard tools, such as the Microsoft compilers mentioned above, while the target code is built using TI's toolset, hosted on the Windows platform. These two types of code interact with each other via the NT driver. However, only a single-threaded task can be run on each DSP using the basic tools. This contrasts with our previous (V)8 RES, whose (Transputer) processor ran several simultaneous tasks (e.g., for communication with the host PC at low priority while servicing the real-time hardware at high priority). In order to achieve this multi-tasking capability on the target DSPs, we have selected a Real-Time Operating System (RTOS) called Diamond, from the 3L corporation. In fact, Diamond inherits from the same code base as that provided by Inmos for their Transputer, and is thus quite familiar to us in several respects.

Diamond significantly simplifies inter-processor communication between the DSPs and between any DSP and the host, and provides mechanisms for synchronizing these processors. Since our baseline architecture could contain as many as 14 processors (12 DSPs, 1 PowerPC, and 1 Windows host), such facilities may significantly enhance the present RESS capabilities and enable growth for the future.

### **6.3 General Computational Module**

As mentioned above, Dwell Processing is achieved in a PowerPC module; this includes computation of target and clutter angular position relative to the RADAR, and target and clutter spectra based on the frequency and pulse-repetition frequency of the RADAR dwell.

At the present time, we believe that such precomputation (which was handled easily by a single Transputer for the (V)8 RES) can be achieved for as many as six RSG channels by a single PowerPC module (which is at least 100 times faster than the Transputer). Moreover, as this computation needs to occur (as rapidly as possible) only once at the beginning of the dwell, there is no advantage in multitasking this function. Therefore, we have chosen the (simple) Topaz development environment from the PowerPC vendor (Transtech Parallel Systems – a former Transputer vendor), rather than the much more complicated and costly VxWorks RTOS from Wind River Systems. Topaz provides the capability of developing a single-threaded application on the PowerPC target using the Windows development environment, including memory transfers to other processors over the PCI backplane (i.e., to the Windows host or the DSP target memories).

## **7.0 Hardware Application**

### **7.1 Memory Requirement for RSG Real Time Operation**

There are some considerations that have to be addressed in order to keep up with the high processing volume of the RADAR processor. The sizing of hardware resources for each process is generally a tradeoff between processor speed and the amount of fast access storage, which can be made available.

A memory requirement estimate has been developed for a modest real time scenario in the first progress report (attached). Approximately 32 Megabytes of local (RAM) memory is required to address the RADAR example that was presented at the beginning of the report. The DSP modules can be purchased with up to 64 MB of local memory, and so this requirement is not considered a problem. For the non-real-time application of course, there is no memory requirement since the RSG can be run entirely from the mass storage.

### **7.2 RADAR and RSG hardware Preparation**

Since both the RADAR and the RSG are complicated electronic entities, it is essential that prior to their interconnection, that appropriate stand-alone testing be performed to insure that each are operating properly. Listed below in Table 1 are some typical steps that are required to be taken prior to RES testing of the FIREFINDER RADAR. Similar procedures are envisioned for the RSG. In an operator training mode of course, some procedural steps may be skipped depending on the specific requirements.

## **Table 1 Typical FIREFINDER RES TESTING STEPS**

### **PRELIMINARY SETUP**

Preliminary Set-Up of AN/TPQ-36(V)8 RADAR System.

- **Mechanical Set-Up Procedure**
- **Trailer Control Function & Switch Settings.**
- **Hazard and Safety Warnings.**

Preliminary Set-Up of RES

- **Mechanical Set-Up Procedure**
- **Control Function and Switch Setting**

### **VERIFICATION OF RADAR/RES OPERATION**

TPQ-36 (V)8 RADAR/RES Integration

- **(V)8 RES Power Up**
- **TPQ-36(V)8 Power Up**
- **TPQ-36(V)8 Operational Program Load**

Once proper operation of hardware has been ensured, training (or testing) can commence. Supplied below is a summary of the remaining steps that can be required for the RES & Firefinder RADAR test and verification:

### **RADAR/RES TESTING OPERATIONS**

Angel Tests  
Clutter/Target Tests  
Standard Target Tests  
Heavy Loading Tests.  
FAT-1 Tests.  
FAT-3 Tests.

These represent a set of tests that were mutually agreed upon. A similar philosophy will be used on the RSG, where a known configuration of scenarios is used as a gauge... whether it is for operator proficiency or RADAR performance. The tests will be chosen to illuminate shortcomings (or highlights) in the desired aspects.

### **RADAR/RES TEST DATA REDUCTION**

**Band Data Reduction**  
**Compilation of Test Data Results.**

When the scenarios were run and the appropriate Firefinder RADAR performance data was collected, a reduction process provided the desired indicators telling of the health of the RADAR processor and software. The RSG will require a similar process.

### 7.3 Performance Metrics

As noted in Table 1, once RADAR performance data is obtained from the RES test, it is necessary to evaluate results against some reasonable set of metrics. For the training application, for example, the RSG could be used to provide a controlled and repeatable set of testing stimuli against which the performance of different configurations, algorithms, students, etc. could be evaluated.

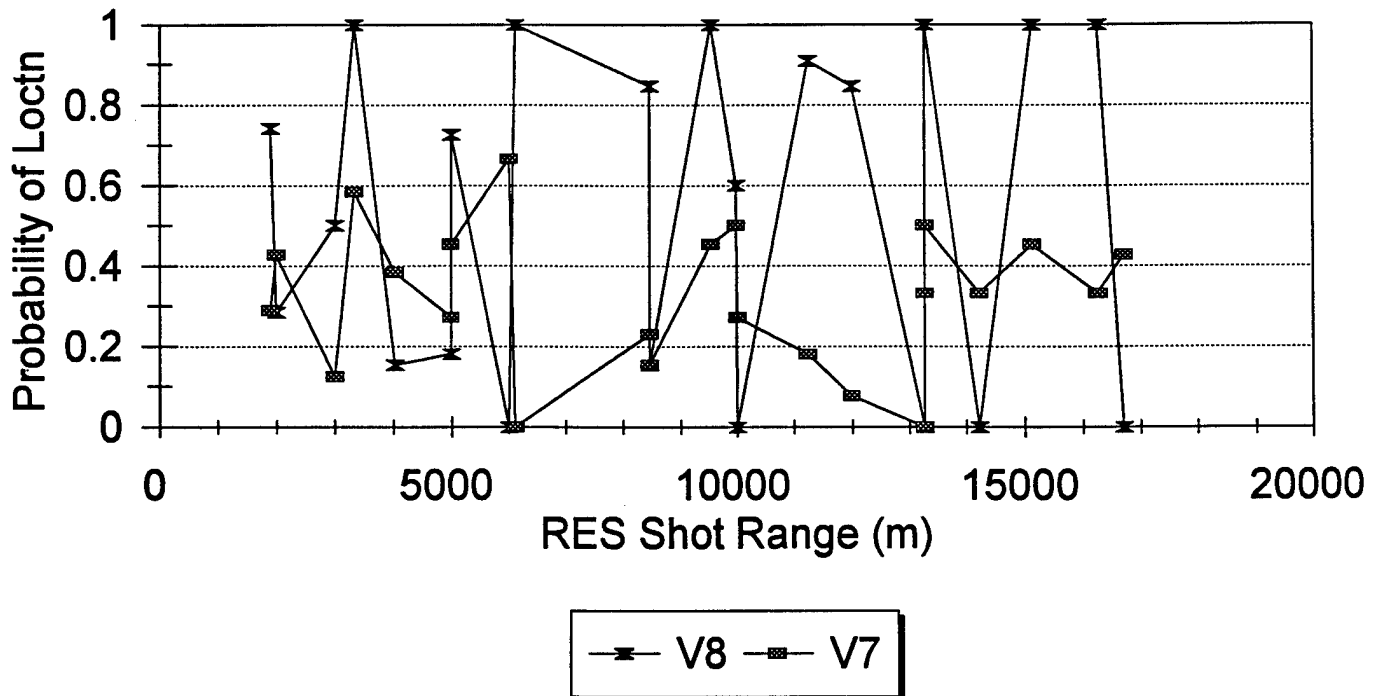
To continue with the example of Table 1, a set of evaluation criteria were developed for the FIREFINDER RADARs, which told of the ability of each RADAR to perform its mission against a fixed set of stressing targets. For these targets, performance data was computed, based on what have become standard measures of Weapon Location RADAR performance.

CEP (Circular Error Probability)- CEP is a measure of the accuracy of a set of RADAR derived RADAR locations for a given shot, in this case as fired repetitively by the FIREFINDER RES. Mathematically, CEP is defined as the radius of a circle, which just encloses 50% of the locations as plotted on an X-Y scatter diagram.

P(l) (Probability of Location)- P(l) is the percentage of (in this case) RES firings which were successfully located by the RADAR.

#### ALR-3 HVY RES TEST RESULTS

(V)8 1/30/96 - (V)7 10/14/93



For both CEP and P(l), it has been found to be useful to plot the performance of a given RADAR as a function of the range to the shots which are "fired" against it. An example

of P(I) results as derived by RES testing at Ft. Sill, OK are shown in the figure, below. Here, the performance of two generations of the AN/TPQ-36 RADAR are compared in terms of a high volume RES generated scenario.

## **7.4 Application to Specific RADAR**

At the time of publication of this report, there had been no target RADAR identified for an RSG application example. The following text describes a generic procedure that would apply to any RADAR.

### **7.4.1 Acquisition of RADAR Documentation**

The RSG can be best matched to a RADAR when the details of the interface are well known. The accuracy of electrical and timing descriptions is extremely important for proper operation of the RADAR-RSG system, and the veracity of RADAR operation in the virtual environment hinges on these. The mechanical interface issues are limited to specific connector and waveguide descriptions and locations of the various connection points.

- 1) Mechanical drawings – of all RF, IF and digital test ports and specific waveguide interfaces (if applicable). These will determine the cable lengths and terminations that the RSG will use to interface to the RADAR.
- 2) Electrical schematics, signal descriptions – A review of the RADAR block diagram will aid in limiting the amount of information that needs to be gathered. Simply put, the interface that allows the RSG to know what modes the RADAR operates in as well as the interface through which the RSG will inject signals back into the RADAR need to be fully described.
- 3) Signal and process timing information – RADAR operational details need to be considered carefully. In modern RADARs, the various sub-assemblies exist as intelligent entities with individual processors and memory. Hence commands are distributed asynchronously through some bus and are queued up until execution time, which is triggered by a timing pulse or time code generator. The RSG needs to accommodate this architecture, associate the proper waveforms with their commands and cue the output in accordance with the same scheme as the RADAR.

The processing details of the RADAR influence the processing horsepower that the RSG needs to carry. Recall that the RSG has a completely scalable DSP card architecture. For example, in a ground based RADAR with relatively low PRF (Pulse Repetition Frequency  $< 3$  KHz) if the command for the antenna position is provided to the antenna sub-assembly 10 uSec prior to the firing of the first beam, but the RADAR uses space fill pulses to determine the wide bandwidth (track) RGC (Receiver Gain Control) values, then the close range targets in the sidelobes need not be represented during the space fill portion of the pulse train (or measurement interval), because they will not contribute to the saturation of the receiver. This significantly reduces the computational



requirement on the DSP engine, which is busy performing the "once per measurement" kinds of tasks. Once these are done (1 mSec typical), the DSP engines can focus on the "per pulse" tasks. In this example, the RADAR processor resources can be sized to handle the "per pulse" tasks with lower fidelity responses provided during the "space fill" portion of the wide bandwidth waveform. Processing the full fidelity signals for each and every pulse would require considerably more processor resources due to the "once per measurement" overhead requirements.

#### **7.4.2 Interface Requirements Generation**

A decision has to be made regarding the best injection point. If only the RADAR algorithms are to be tested or exercised for operator proficiency, then the injection can happen at IF frequencies, or digitally, directly into the RADAR processor. If the receiver hardware is in question, then the injection point has to happen at IF or RF frequencies depending on the exact nature of the problem. The antenna is the only element of the RADAR system that is excluded by the RSG.

Under most circumstances, problems in the receiver hardware can be isolated using available maintenance equipment. Utility as a training tool and as a device to check algorithm development is well addressed by IF injection. This form of injection is a reasonable compromise between the wiring complexity of digital injection and the complexity associated with high fidelity RF conversion.

#### **7.4.3 Figure of Merit**

A "level of goodness" needs to be quantified. This can vary significantly depending on the RSG application. For example, as a training tool, the figure of merit for the RSG scenario performance can simply be the number of targets that are successfully registered and reported by the operator. As a RADAR performance evaluation tool, the pass/fail criteria will have to include quantities such as "#targets correctly identified / total # targets", position accuracy as a function of range, as a function of target type, or, incidence of false alarms as a function of time, position, clutter level or ECM etc.

The utility of the RSG can be best exploited by intelligent choice of the figure of merit used to grade the RADAR-RSG-operator interaction.

#### **7.4.4 Scenario Generation**

An associated issue is the collection of a scenario database. The realism of the test and/or the training session is primarily dependent on this database. For example, having accurate data of the terrain, the weather, any vegetation on the terrain, the targets and the EMI environment for a desired RADAR emplacement can provide invaluable training and performance indication.

If the database is accurate, then the number of required RADARs to perform a desired mission can be accurately predicted as well. Also, the operator proficiency required to achieve

the desired goals can be enhanced by off-site simulation training or by assessing the optimal work/rest schedule for the engagement, depending on the operator's performance during the simulation trial period.

## **8.0 Running the RSG as a Desktop Simulator**

A powerful additional capability may be obtained from the RSG by adapting it to run in Standalone mode as a "Desktop Simulator". If the RSG is look at in terms of signal generation capabilities, it provides a high fidelity means of generating non-real time RADAR signals. Thus, the data generation features of the RSG may be directly applied to drive simulated modules of the candidate RADAR itself. And, since the simulation need not run in real time to keep up with the actual RADAR hardware, it is possible add data monitor and recording modules to provide valuable insight as to RADAR performance before the equipment is actually built and tested.

Our experience with predecessor RES units has shown that running the RSG in this manner provides an excellent tool to checkout the RES implementation before it is connected to the actual RADAR. Thus, the RSG would be "pre-integrated" with a simulated RADAR before it is delivered, making the field integration process much more efficient.

The actual modifications to the RSG to allow it to function as a Desktop Simulator are quite straightforward.

1. Develop a simple model for the RADAR and its control signals which operate under computer clocking rather than the RADAR hardware.
2. "Stub out" the real time I/O functions on the RSG and replace them with suitable control and display functions which will enable control of the Desktop Simulator and display of its results
3. All of the RSG databases, models, signal generation functions and control signals can be directly applied to the Desktop Simulator
4. The exact nature of the signal interface may be as simple as range ordered RADAR signal definitions for each target to actual range bin I/Q data which can be used to drive simulations of the processing hardware.

## **9.0 Conclusions and Summary**

This Final Report has summarized work performed on the Multispectral High Fidelity RADAR Scene Generator SBIR (Phase I), Contract # N68355-99-C-0126.

The product approach developed herein has excellent potential as a training tool for various RADARs. It provides an operator interface with the actual on site RADAR hardware. There is some minimum modification that is done to the RADAR system to accept the RSG. Due to the flexibility of the RSG output, RADAR signal injection can be made anywhere from

the RF front end (after the 1<sup>st</sup> LNA), through various points in the IF processing chain, all the way to selected points in the digital processing.

Additionally, The RSG serves as a performance verification tool for RADAR hardware as it lives out its life cycle. As the hardware or operational software receives upgrades, the RSG can be applied as a final checkout prior to live engagement. The benefit of using the RSG is that the RADAR environment is electronically synthesized, and therefore is exactly repeatable. The RADAR performance improvement then can be assessed objectively as the RADAR goes through its upgrade cycles.

The Multispectral High Fidelity RADAR Scene Generator represents a portable and comprehensive test and training environment for the Navy's RADAR assets. Additionally, the option of providing a RADAR evaluation service for other RADAR sponsors becomes available, where the Navy can maintain a center that could assist other RADAR users in the evaluation of RADAR assets and training of operators.

## **10.0 Plans for Follow on Phase**

A specific RADAR application is anticipated for the follow on phase. Phase II for the Multispectral RSG will address the following issues:

Building a generic RSG – A real time version of the RSG hardware described herein will be built.

Interface definition for RADAR under consideration – Electrical and mechanical interfaces will be developed for the specific RADAR through which the utility of the RSG will be demonstrated.

Candidate CDRL for Phase II effort – There are a number of data items that will be provided with the RSG:

Database – this item will define all the components that were used in deriving the database for the RSG. The references used and the pertinent data from each reference will be traced to the model.

Mechanical Drawings - Mechanical drawings will be provided (in contractor's format) describing the construction of the RSG hardware. Descriptions for all parts used in the RSG will be provided.

Schematics – Electrical interconnections will be provided for all the COTS equipment within the RSG, and detailed schematics will be provided for any electronic components designed and built by Malibu Research.

Manuals – the following manuals will be provided as hardcopy as well as in "PDF" format on CD ROM media. The operator will be able to access this information while at the RSG operator's console.

Operator's manual – An operator's manual describing initialization, calibration and operation of the RSG will be developed during the second phase of the SBIR.

Maintenance/Troubleshooting manual – Instructions to effectively use the delivered schematics and mechanical drawings will be provided in manual form.

A detailed breakdown of tasks will be provided when the SBIR phase II proposal is submitted.

## **11.0 Schedule**

The detailed schedule of tasks and milestones will be submitted with the phase II proposal as well.